

LISTING OF THE CLAIMS:

1. (Currently Amended): A method of mutual process monitoring for detecting a termination of a process within a plurality of processes in a data processing system, the method comprising the steps of:

establishing, ~~within the plurality of processes,~~ a monitoring policy, wherein the monitoring policy assigns a ~~first~~ each process within the plurality of processes to monitor a ~~second~~ one other process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

responsive to a termination of execution of the ~~second~~ a monitored process, determining the cause of the execution termination by the ~~[[first]]~~ monitoring process; and

responsive to a determination that the ~~second~~ monitored process terminated execution in an abnormal manner, attempting to restart the ~~second~~ monitored process by the ~~[[first]]~~ monitoring process.

2. (Currently Amended): The method as recited in claim 1, further comprising: opening a communications link between the ~~[[first]]~~ monitoring process and the ~~second~~ monitored process; and

posting a blocking read by the ~~[[first]]~~ monitoring process in order to detect termination of the ~~second~~ monitored process.

3. (Original): The method as recited in claim 2, wherein the communications link is a watch FIFO (first-in first-out) communications link.

4. (Original): The method as recited in claim 3, wherein posting a blocking read to the watch FIFO communications link is performed in a single thread of execution.

5. (Currently Amended): The method as recited in claim 2, wherein the [[first]] monitoring process posts a blocking operation on the communications link that completes the blocking operation on termination of the ~~second~~ monitored process.

6. (Currently Amended): The method as recited in claim 2, wherein the blocking operation is performed in a separate execution thread within the [[first]] monitoring process in order to allow normal process operation of the ~~second~~ monitored process to continue.

7. (Currently Amended): The method as recited in claim 1, further comprising:
responsive to a determination that the ~~second~~ monitored process terminated execution in a normal manner, modifying the monitoring policy, wherein the monitoring policy is modified to assign the [[first]] monitoring process to monitor a [[third]] different process within the plurality of processes; and
monitoring of the [[third]] different process by the [[first]] monitoring process for termination of execution.

8. (Canceled)

9. (Currently Amended): A method for inserting [[a]] an additional process within a plurality of processes containing a ~~first process and a~~ mutually monitoring process policy in a data processing system, the method comprising the steps of:

establishing a monitoring policy, wherein the monitoring policy assigns each process within the plurality of processes to monitor one other process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

receiving a request for a ~~second~~ an additional process to join the plurality of processes;

responsive to the ~~second~~ additional process joining the plurality of processes, randomly selecting ~~the first~~ a process within the plurality of processes to monitor the ~~second~~ additional process; and

modifying the mutually monitoring process policy, wherein the selected ~~[[first]]~~ process monitors the ~~second~~ additional process for termination of execution.

10. (Currently Amended): The method as recited in claim 9, wherein the ~~second~~ additional process monitors the process previously monitored by the selected ~~[[first]]~~ process.

11. (Currently Amended): The method as recited in claim 9, wherein the ~~second~~ additional process joining the plurality of processes joins as a child of the ~~[[first]]~~ selected process.

12. (Original): The method as recited in claim 9, wherein the request is received by a inter-process communications (IPC) mechanism.

13. (Canceled)

14. (Original): The method as recited in claim 9, further comprising:
locking a state file in a join communications link; and
blocking any other processes attempting to simultaneously join the plurality of processes.

15. (Original): The method as recited in claim 14, wherein the communications link is a first-in first-out communications link.

16. (Original): The method as recited in claim 14, wherein blocking any other processes attempting to simultaneously join the plurality of processes is by a file lock.

17. (Currently Amended): A data processing system for mutual process monitoring for detecting a termination of a process within a plurality of processes in a data processing system, comprising:

establishing means for establishing, ~~within the plurality of processes,~~ a monitoring policy, wherein the monitoring policy assigns a ~~first~~ each process within the plurality of processes to monitor a ~~second one other~~ process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

determining means, responsive to a termination of execution of ~~the second~~ a monitored process, for determining the cause of the execution termination by the [[first]] monitoring process; and

attempting means, responsive to a determination that the ~~second~~ monitored process terminated execution in an abnormal manner, for attempting to restart the ~~second~~ monitored process by the [[first]] monitoring process.

18. (Currently Amended): The data processing system as recited in claim 17, further comprising:

opening means for opening a communications link between the [[first]] monitoring process and the ~~second monitored~~ process; and

posting means for posting a blocking read by the [[first]] monitoring process in order to detect termination of the ~~second monitored~~ process.

19. (Original): The data processing system as recited in claim 18, wherein the communications link is a watch FIFO (first-in first-out) communications link.

20. (Original): The data processing system as recited in claim 19, wherein posting a blocking read to the watch FIFO communications link is performed in a single thread of execution.

21. (Currently Amended): The data processing system as recited in claim 18, wherein the [[first]] monitoring process posts a blocking operation on the communications link that completes the blocking operation on termination of the ~~second~~ monitored process.

22. (Currently Amended): The data processing system as recited in claim 18, wherein the blocking operation is performed in a separate execution thread within the [[first]] monitoring process in order to allow normal process operation of the ~~second~~ monitored process to continue.

23. (Currently Amended): The data processing system as recited in claim 17, further comprising:

modifying means, responsive to a determination that the ~~second~~ monitored process terminated execution in a normal manner, for modifying the monitoring policy, wherein the monitoring policy is modified to assign the [[first]] monitoring process to monitor a [[third]] different process within the plurality of processes; and

monitoring means for monitoring of the [[third]] different process by the [[first]] monitoring process for termination of execution.

24. (Canceled)

25. (Currently Amended): A data processing system for inserting [[a]] an additional process within a plurality of processes containing a ~~first process and a~~ mutually monitoring process policy in a data processing system, the method comprising the steps of:

establishing means for establishing a monitoring policy, wherein the monitoring policy assigns each process within the plurality of processes to monitor one other process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

receiving means for receiving a request for a ~~second~~ an additional process to join the plurality of processes;

selecting means, responsive to the ~~second~~ additional process joining the plurality of processes, for randomly selecting ~~the first~~ a process within the plurality of processes to monitor the ~~second~~ additional process; and

modifying means for modifying the mutually monitoring process policy, wherein the selected ~~[[first]]~~ process monitors the ~~second~~ additional process for termination of execution.

26. (Currently Amended): The data processing system as recited in claim 25, wherein the ~~second~~ additional process monitors the process previously monitored by the selected ~~[[first]]~~ process.

27. (Currently Amended): The data processing system as recited in claim 25, wherein the ~~second~~ additional process joining the plurality of processes joins as a child of the ~~[[first]]~~ selected process.

28. (Original): The data processing system as recited in claim 25, wherein the request is received by a inter-process communications (IPC) mechanism.

29. (Canceled)

30. (Original): The data processing system as recited in claim 25, further comprising:
locking means for locking a file used to join the communications link; and
blocking means for blocking any other processes attempting to simultaneously join the plurality of processes.

31. (Original): The data processing system as recited in claim 30, wherein the communications link is a FIFO communications link.

32. (Original): The data processing system as recited in claim 30, wherein blocking any other processes attempting to simultaneously the plurality of processes is by a file lock.

33. (Currently Amended): A data processing system, comprising:
- a bus system;
 - a memory, containing a set of instructions and ~~a first and second processes within~~ a plurality of processes, connected to the bus system; and
 - a processing unit connected to the bus system, wherein the processing unit executes the set of instructions from the memory to establish, ~~within the plurality of processes,~~ a monitoring policy, wherein the monitoring policy assigns ~~the first each~~ process within the plurality of processes to monitor ~~the second one other~~ process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system, responsive to a termination of execution of ~~the second a~~ monitored process, the processing unit instructs the ~~[[first]]~~ monitoring process to determine the cause of the execution termination, and responsive to a determination that the ~~second monitored~~ process terminated execution in an abnormal manner, the processing unit instructs the ~~[[first]]~~ monitoring process to attempt to restart the ~~second monitored~~ monitored process.
34. (Currently Amended): A data processing system, comprising:
- a bus system;
 - a memory, containing a set of instructions and ~~a first process within~~ a plurality of processes ~~and a second process,~~ connected to the bus system; and
 - a processing unit connected to the bus system, wherein the processing unit executes the set of instructions from the memory to establish a monitoring policy, wherein the monitoring policy assigns each process within the plurality of processes to monitor one other process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system, receive a request for the ~~second an additional~~ process to join the plurality of processes, ~~responsive~~ respond to the ~~second additional~~ process joining the plurality of processes, the processing unit randomly selects ~~the first a~~ process within the plurality of processes to monitor the ~~second additional~~ additional process, and the processing unit instructs the ~~[[first]]~~ selected process to

modify the mutually monitoring process policy, wherein the selected [[first]] process monitors the ~~second~~ additional process for termination of execution.

35. (Currently Amended): A computer program product in a computer-readable medium for mutual process monitoring for detecting a termination of a process within a plurality of processes in a data processing system, comprising:

instructions for establishing, ~~within the plurality of processes,~~ a monitoring policy, wherein the monitoring policy assigns a ~~first~~ each process within the plurality of processes to monitor ~~a second one other~~ process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

instructions, responsive to a termination of execution of ~~the second~~ monitored process, for determining the cause of the execution termination by the [[first]] monitoring process; and

instructions, responsive to a determination that the ~~second~~ monitored process terminated execution in an abnormal manner, for attempting to restart the ~~second~~ monitored process by the [[first]] monitoring process.

36. (Currently Amended): The computer program product as recited in claim 35, further comprising:

instructions for opening a communications link between the [[first]] monitoring process and the ~~second~~ monitored process; and

instructions for posting a blocking read by the [[first]] monitoring process in order to detect termination of the ~~second~~ monitored process.

37. (Original): The computer program product as recited in claim 36, wherein the communications link is a watch FIFO communications link.

38. (Original): The computer program product as recited in claim 37, wherein the instructions for posting a blocking read to the watch FIFO communications link is performed in a single thread of execution.

39. (Currently Amended): The computer program product as recited in claim 36, wherein the [[first]] monitoring process posts a blocking operation on the communications link that completes the blocking operation on termination of the ~~second~~ monitored process.

40. (Currently Amended): The computer program product as recited in claim 36, wherein the blocking operation is performed in a separate execution thread within the [[first]] monitoring process in order to allow normal process operation of the ~~second~~ monitored process to continue.

41. (Currently Amended): The computer program product as recited in claim 35, further comprising:

instructions, responsive to a determination that the ~~second~~ monitored process terminated execution in a normal manner, for modifying the monitoring policy, wherein the monitoring policy is modified to assign the [[first]] monitoring process to monitor a [[third]] different process within the plurality of processes; and

instructions for monitoring of the [[third]] different process by the [[first]] monitoring process for termination of execution.

42. (Canceled)

43. (Currently Amended): A computer program product in a computer-readable medium for inserting [[a]] an additional process within a plurality of processes containing a ~~first process and a~~ mutually monitoring process policy in a data processing system, comprising:

instructions for establishing a monitoring policy, wherein the monitoring policy assigns each process within the plurality of processes to monitor one other process within the plurality of processes, thereby forming a ring of mutually monitoring processes within the data processing system, wherein the data processing system is a single node within a multi-node system;

instructions for receiving a request for a ~~second~~ an additional process to join the plurality of processes;

instructions, responsive to the ~~second~~ additional process joining the plurality of processes, for randomly selecting ~~the first~~ a process within the plurality of processes to monitor the ~~second~~ additional process; and

instructions for modifying the mutually monitoring process policy, wherein the selected ~~[[first]]~~ process monitors the ~~second~~ additional process for termination of execution.

44. (Currently Amended): The computer program product as recited in claim 43, wherein the ~~second~~ additional process monitors the process previously monitored by the selected ~~[[first]]~~ process.

45. (Currently Amended): The computer program product as recited in claim 43, wherein the ~~second~~ additional process joining the plurality of processes joins as a child of the ~~[[first]]~~ selected process.

46. (Original): The computer program product as recited in claim 43, wherein the request is received by a inter-process communications (IPC) mechanism.

47. (Canceled)

48. (Original): The computer program product as recited in claim 43, further comprising:

instructions for locking a state file in a join first-in first-out communications link (FIFO); and

instructions for blocking any other processes attempting to simultaneously join the plurality of processes.

49. (Original): The computer program product as recited in claim 48, wherein the communications link is a FIFO communications link.

50. (Original): The method as recited in claim 48, wherein blocking any other processes attempting to simultaneously join the plurality of processes is by a file lock.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.